Biomedical Signal Processing

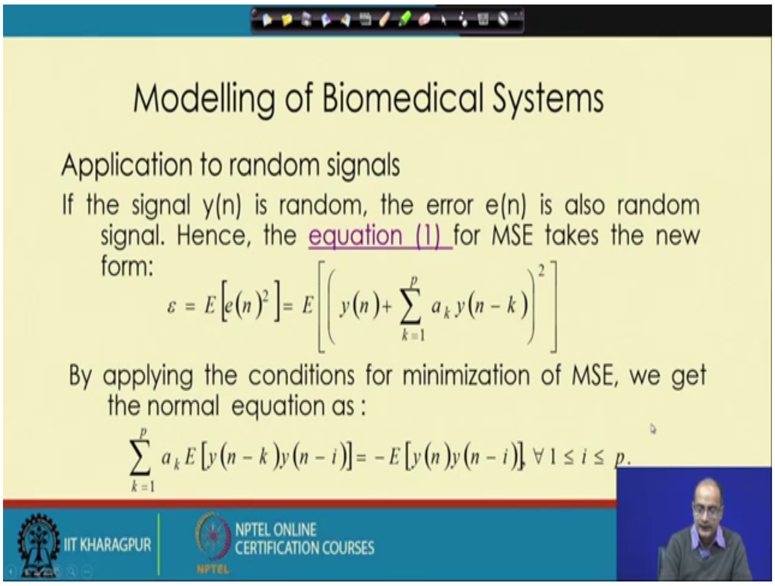
Prof. Sudipta Mukhopadhyay
Department of Electrical and Electronics Communication Engineering
Indian Institute of Technology, Kharagpur

Lecture - 43
Modelling of Biomedical Systems (Contd.)

In the previous module when we are talking about AR model and showing that how we can find out the coefficients, strictly speaking it is not actually AR model it is just a linear prediction model. The reason is we were looking at only a deterministic signal, we have not considered it as a random signal, we have just taken that signal is a deterministic one and trying to find out a linear prediction model which fits there.

Now, let us take the case that if the signal is a random signal how we proceed about it and, to be precise that AR model is defined only for the assumption that signal is random because, the driving signal is a white noise which is itself a random process ok.

(Refer Slide Time: 01:23)



Modelling of Biomedical Systems

Application to random signals

If the signal y(n) is random, the error e(n) is also random signal. Hence, the equation (1) for MSE takes the new form:

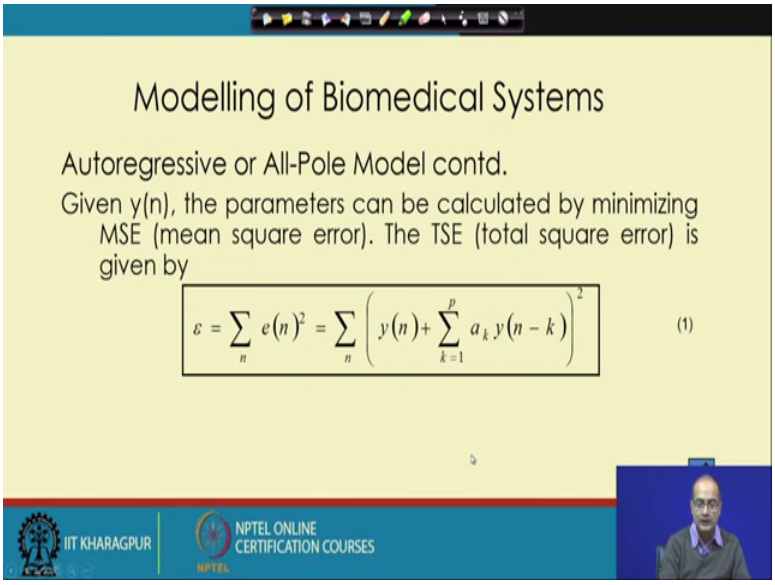
$$\varepsilon = E\left[e(n)^2\right] = E\left[\left(y(n) + \sum_{k=1}^{p} a_k y(n-k)\right)^2\right]$$

By applying the conditions for minimization of MSE, we get the normal equation as :

$$\sum_{k=1}^{p} a_k E\left[y(n-k)y(n-i)\right] = -E\left[y(n)y(n-i)\right], \forall 1 \le i \le p.$$

So, this difference please keep in mind and, here if we take the y n is a random actually signal. So, error en which is the prediction error of y n is also random; So, in this case the equation 1, that equation 1.

(Refer Slide Time: 01:57)



Modelling of Biomedical Systems

Autoregressive or All-Pole Model contd.

Given y(n), the parameters can be calculated by minimizing MSE (mean square error). The TSE (total square error) is given by

$$\varepsilon = \sum_{n} e(n)^2 = \sum_{n} \left(y(n) + \sum_{k=1}^{p} a_k y(n-k) \right)^2 \qquad (1)$$

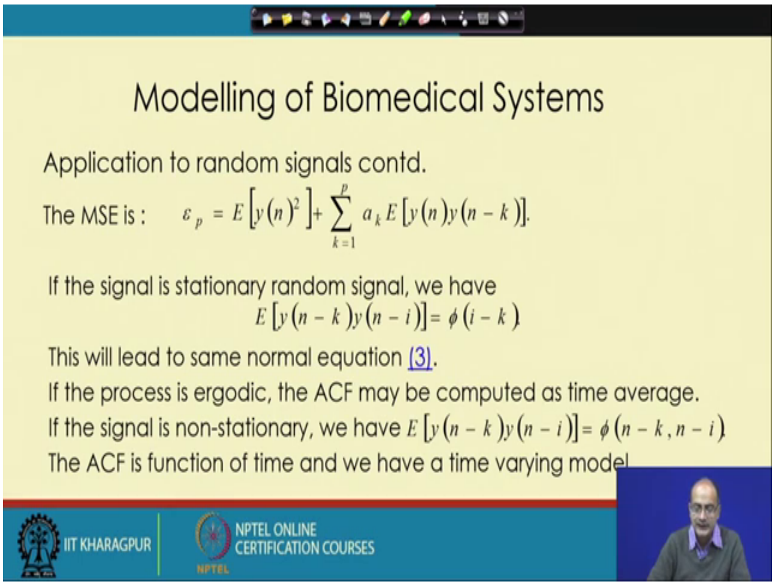
What we had for the total square error, or a scale version of a mean square error it would take a new form ok, what we have done instead of the summation, we have used the expectation operator this is the change. Now, the natural question could be could be that why we are doing all such tricks, the point is that for the beauty of the this parametric model is there that when we know that this is the model, we try to find out the model parameters out of the signal, we call for a random process that these values are just in symbol.

So, once we have the outputs y n out of that we find out the model parameters. And once we get the model parameters, we can characterise that process in terms of model parameters, we do not have much need of those exact values of y n which are so, important in nonparametric methods.

Once we get those model parameters, or the model itself we can say that we can forget about the signal we know signal is nothing, but it is an output at a at a point of time it came out of this model, at a different point of time another set of value may come out of that model and as you know the model we can do all the work with the help of that modern. So, the signal becomes of less importance after the model is derived and, once we have that all the parameters in our hand ok. So, that is the beauty of the parametric model it gives is a compact representation of the whole thing.

Now, here let us come back that we have the new definition of the mean square error using that again, we go in our known way of finding out the minimum error. So, using that minimum error criteria; that means, taking the partial derivative with respect to a ks and equating that partial derivative to 0, we get a set of normal equations that is consisting of a ks and expectation of the that second order terms of y primarily of the form y n minus k y sn minus i in this form.

(Refer Slide Time: 05:13)



Modelling of Biomedical Systems

Application to random signals contd.

The MSE is : $\varepsilon_p = E\left[y(n)^2\right] + \sum_{k=1}^{p} a_k E\left[y(n)y(n-k)\right].$

If the signal is stationary random signal, we have

$$E\left[y(n-k)y(n-i)\right] = \phi(i-k)$$

This will lead to same normal equation (3).

If the process is ergodic, the ACF may be computed as time average.

If the signal is non-stationary, we have $E\left[y(n-k)y(n-i)\right] = \phi(n-k, n-i)$

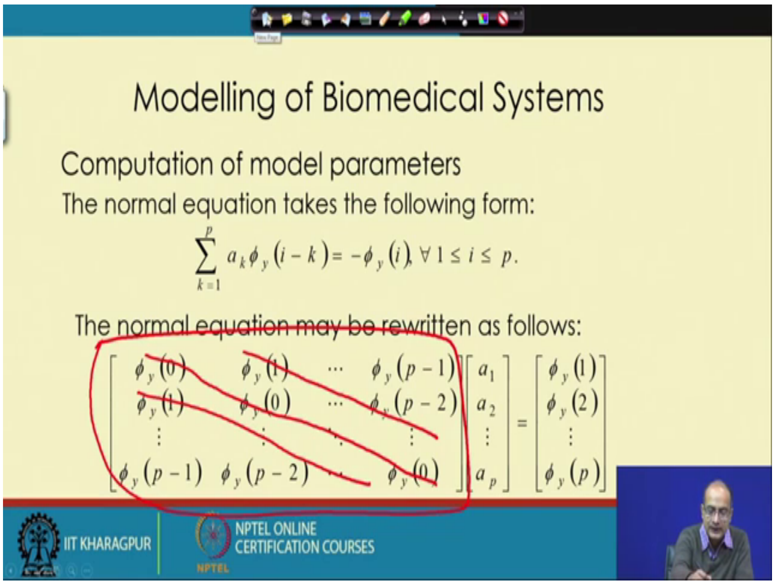
The ACF is function of time and we have a time varying model

IIT KHARAGPUR

NPTEL ONLINE CERTIFICATION COURSES

So, if we solve that we can get the value of a ks and, again with this expected values we can actually get the mean square error value also with the help of the expected values of y n minus i into y n minus k and a ks. Now if the signal is a stationary random process, the term that y n minus k y n minus I, or rather the expectation of this product, we can represented as the autocorrelation of lag i minus k and, we get actually the same form as previous that equation number 3.

(Refer Slide Time: 06:13)



Modelling of Biomedical Systems

Computation of model parameters

The normal equation takes the following form:

$$\sum_{k=1}^{p} a_k \phi_y(i-k) = -\phi_y(i), \forall 1 \le i \le p.$$

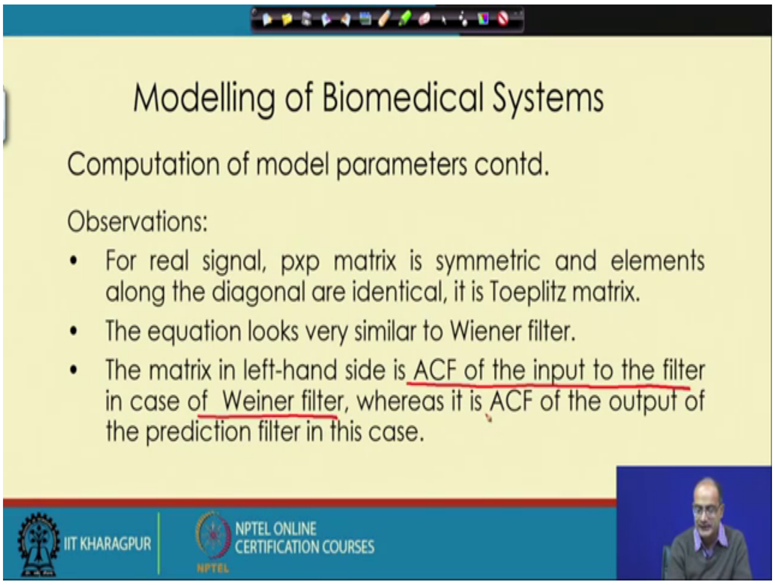
The normal equation may be rewritten as follows:

$$\begin{bmatrix} \phi_y(0) & \phi_y(1) & \cdots & \phi_y(p-1) \\ \phi_y(1) & \phi_y(0) & \cdots & \phi_y(p-2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_y(p-1) & \phi_y(p-2) & \cdots & \phi_y(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} \phi_y(1) \\ \phi_y(2) \\ \vdots \\ \phi_y(p) \end{bmatrix}$$

That the equation number 3 which give us those equations; So, we are getting the same set of normal equations with phi and, here we get a particular characteristics that these matrix is something very interesting. If you look at that all the diagonal terms the principal diagonal it is with 5 0 and you will get that, that half diagonal terms also they are having the same values here, it is phi 1 it is phi actually minus 1 mean again phi one for real signal ok.

So, it has not only the symmetry all the diagonal terms, they are actually having the same value.

(Refer Slide Time: 07:34)



Modelling of Biomedical Systems

Computation of model parameters contd.

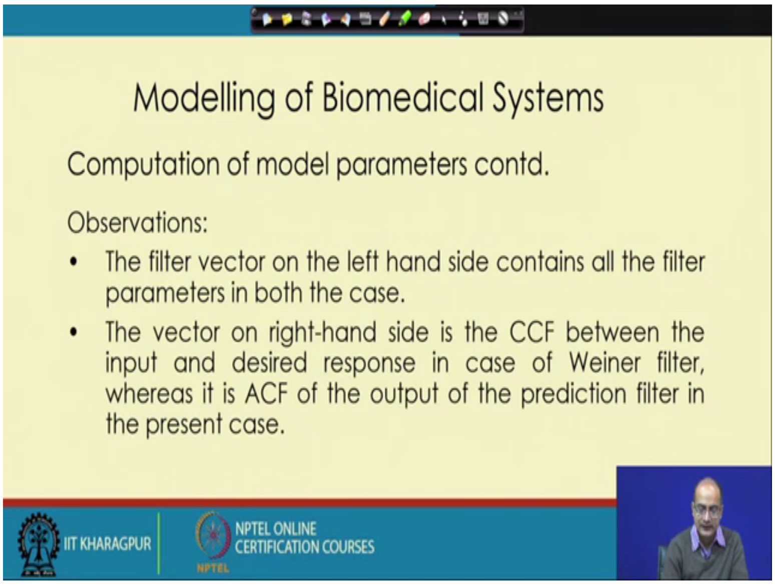
Observations:

- For real signal, pxp matrix is symmetric and elements along the diagonal are identical, it is Toeplitz matrix.
- The equation looks very similar to Wiener filter.
- The matrix in left-hand side is ACF of the input to the filter in case of Weiner filter, whereas it is ACF of the output of the prediction filter in this case.

So, for the real signal this p by p matrix, it is symmetric and the elements along the diagonals are identical. And such matrix is given a special name as Toeplitz matrix ok. So, this Toeplitz matrix this is something very special structure and, this equation what we got it look very similar to the earlier wiener filter that, Weiner filter also implemented using a tabular filter here also we have the same structure and same equation; however, there are certain small differences appear.

In this case that ACF what we is used in the left hand side was the ACF of the input in case of the Weiner filter whereas, in this case it is a ACF of the output of the prediction filter. So, this is the first difference that in the left hand side, the coefficients or the terms what we get though both the case it is ACF, in case of Weiner filter it was ACF of the input and in this case the ACF of the prediction filter that is; that means, the that of the output signal.

(Refer Slide Time: 09:28)



Modelling of Biomedical Systems

Computation of model parameters contd.

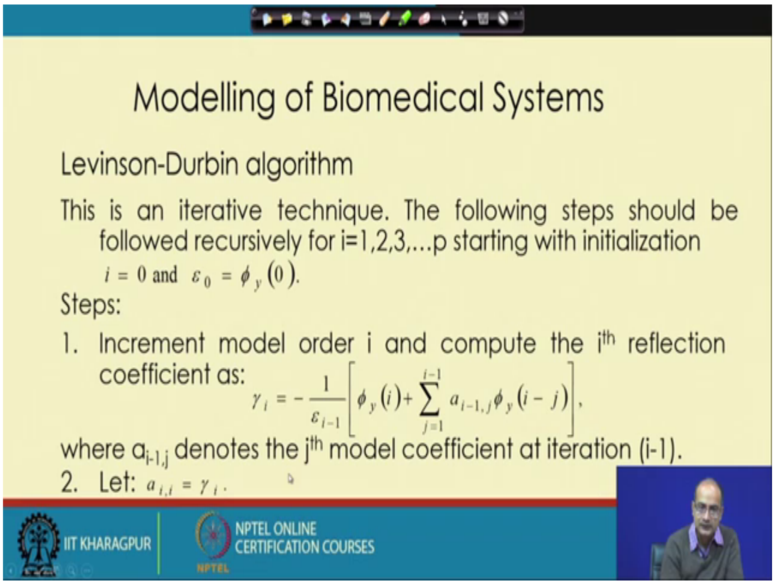
Observations:

- The filter vector on the left hand side contains all the filter parameters in both the case.
- The vector on right-hand side is the CCF between the input and desired response in case of Weiner filter, whereas it is ACF of the output of the prediction filter in the present case.

The filter vector on the left hand side contents all the filter parameters in both the case. So, what we try to derive, the filter parameters if both the case they are in the left hand side.

Next we look at the right hand side and for the right hand side what we get that we have the cross correlation between the input and the desired response in case of Weiner filter, whereas for the air model we have again the ACF of the output of the prediction filter; that means, that of y ok. So, that is the primary difference though, we look similar because in the left hand side, we have autocorrelation and we have all the parameters they are not exactly the same.

(Refer Slide Time: 10:40)



Modelling of Biomedical Systems

Levinson-Durbin algorithm

This is an iterative technique. The following steps should be followed recursively for i=1,2,3,…p starting with initialization $i = 0$ and $\varepsilon_0 = \phi_y(0)$.

Steps:

1. Increment model order i and compute the i^{th} reflection coefficient as: $$\gamma_i = -\frac{1}{\varepsilon_{i-1}}\left[\phi_y(i) + \sum_{j=1}^{i-1} a_{i-1,j}\phi_y(i-j)\right],$$

where $a_{i-1,j}$ denotes the j^{th} model coefficient at iteration (i-1).

2. Let: $a_{i,i} = \gamma_i$.

Now, to take actually the advantage of the special structure of this matrix which is a Toeplitz matrix, we can make use of the Levinson Durbin, Durbin algorithm it is an iterative technique which has different benefits the first benefits is it helps to reduce the complexity. The complexity of the computation or the solution of the that the set of linear equations, we know that to compute directly what we can do, we can take the inverse of that matrix and, taken in the right hand side we can multiply both the sides by the inverse of that autocorrelation matrix.

So, left hand side will have those filter parameters right hand side will have the inverse of the autocorrelation matrix multiplied by the vector of autocorrelation and, then we can get those filter parameters after the multiplication in the right hand side.

Now, the amount of computation what we have to do, we can reduce that using the special structure of the Levinson Durbin algorithm. So, first let us see that how the process is defined for that here the steps are given the first step is that, we have to do the initialisation and initialisation first the index is 0 and for that the error e 0 it is nothing, but the signal autocorrelation at the 0th lag.

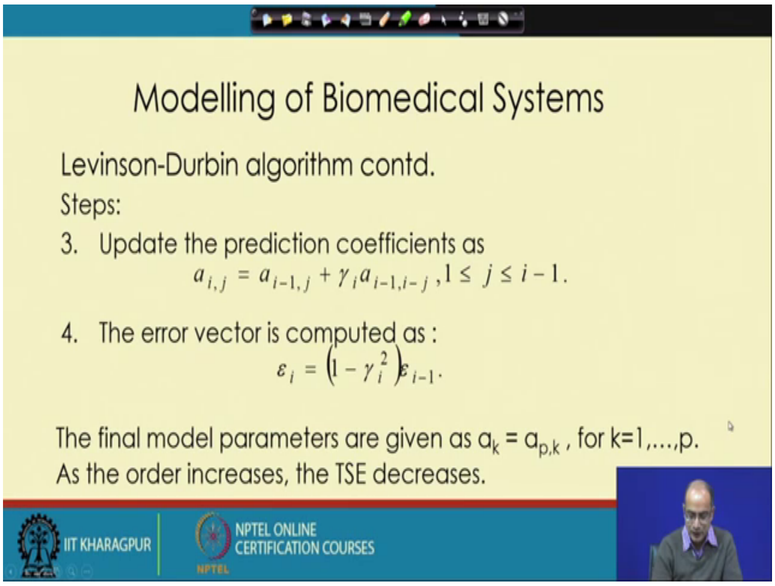
In other word when the prediction order is 0 then prediction would be 0. So, the prediction error is nothing, but same as the that whole single. So, the energy of that would be same as the signal energy and, now we have to go through number of iteration have to be more precise for p iterations ok.

So, let us go through those steps, the first point is we need to increment the model order and compute the i-th reflection coefficient. Now if you look at the form of the reflection coefficient you would get that, you have seen this form this form is nothing, but the error of a actually i-th order model that, the i-th order model what would be the prediction error, we are actually using that. And then it is scaled by the previous the error at the previous model order that is eta i minus 1 ok.

So, it is a ratio of the two errors we can say. So, this is the reflection coefficient and after we compute the reflection coefficient that, we have to go for that that the next step for that actually we need to make use of that computation of that a i minus 1 comma j, which suggest that j-th model coefficient at the iteration i minus 1; that means, to the previous level what was the that the coefficients of the a i prediction coefficients, we make use of that and find out that what is the error and, then we get the the model that reflection coefficient.

Now, we will update actually all the that prediction coefficient, the new 1 a i i that up to that the previous 1, it was i value was i minus 1. So, that i a i comma i is new, which is appearing first time that is taken as same as the reflection coefficient.

(Refer Slide Time: 15:53)



Modelling of Biomedical Systems

Levinson-Durbin algorithm contd.

Steps:

3. Update the prediction coefficients as
$$a_{i,j} = a_{i-1,j} + \gamma_i a_{i-1,i-j}, 1 \le j \le i-1.$$

4. The error vector is computed as :
$$\varepsilon_i = \left(1 - \gamma_i^2\right)\varepsilon_{i-1}.$$

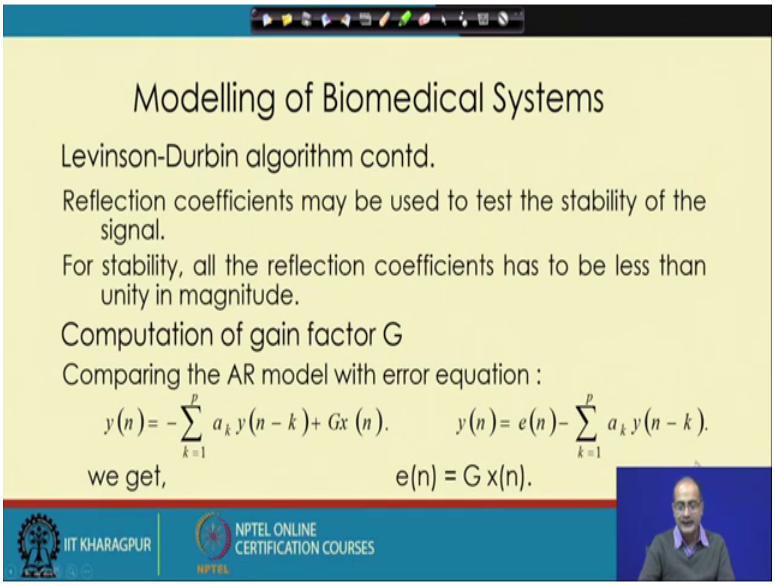
The final model parameters are given as $a_k = a_{p,k}$, for k=1,...,p.

As the order increases, the TSE decreases.

And the other coefficients which were there earlier they are updated, those values represented as a i comma j for j is in between 1 to i minus 1, we take what was the previous value in the last iteration into that reflection coefficient into that, we get a reverse actually a flip kind of thing of the coefficients in the previous iteration a flip version of it.

So, this is the way the updation happens and, we can compute the error vector eta i it is nothing, but the previous error into 1 minus square of the reflection coefficient. So, these steps actually are followed p times and at the end of it we get the final model parameters a ks, they are given as that at the end of the p iteration coefficient value ok. So, what we are looking at how to find out the filter parameters that we get by the that p iterations here. Now as the model order increase this total squared error TSE will decrease.

(Refer Slide Time: 18:00)



Modelling of Biomedical Systems

Levinson-Durbin algorithm contd.

Reflection coefficients may be used to test the stability of the signal.

For stability, all the reflection coefficients has to be less than unity in magnitude.

Computation of gain factor G

Comparing the AR model with error equation :

$$y(n) = -\sum_{k=1}^{p} a_k y(n-k) + Gx(n). \qquad y(n) = e(n) - \sum_{k=1}^{p} a_k y(n-k).$$

we get, e(n) = G x(n).

Now, let us look at couple of interesting thing that from the book, we have taken these points book of professor Rangayyan, it is written that reflection coefficient may be used to test the stability of the signal. And the next point is for stability all the reflection coefficient has to be less than unity in magnitude, I have not change those statements because, they are in the book and if you are following that book you may be a bit ruffled if I change them.

But the fact is if you go through the SNKS book, you will get that reflection coefficients have a nice property that they would always be less than unity in magnitude ok, that you can actually convince yourself, you can take an example and compute them, or you can just look at that equations, let us go back to that how reflection coefficients are computed.

Here if you look at this equation of reflection coefficient, it is nothing but actually the error, in the previous that the present error divided by the previous error. So, as it is we know we increase the model order, this error decreases this value cannot be more than 1, it is by design it would be less than 1. In fact, that is one of the beauty of the Levinson Durbin algorithm, the filter coefficient it will generate in this way it will make sure that the filter would be stable because, they would be reflected inside the unit circle.

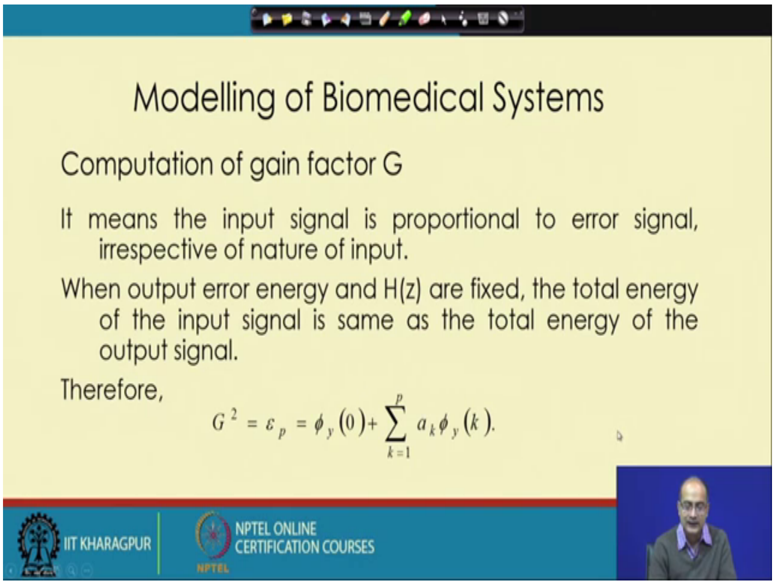
Earlier we are we have gone through the concept of minimum phase signal, where all the poles and 0s, they are reflected inside here Levinson Durbin algorithm using that reflection coefficient, they are achieving that property they are reflecting all the poles here because here we are not dealing with 0, inside the unit circle and all those AR parameters, they are corresponding to the minimum phase signal, or to put in a more simpler way that we always get a stable signal.

So, these two lines they are not precisely correct they are they should be interpreted in a different way that Levinson Durbin algorithm, that gives stable filter and reflection coefficients make sure that their values are magnitude would be less than unity and, thereby they are actually reflected inside. So, all the coefficients what we get, we could make use of it without any problem of stability ok.

So, now let us we can look at how to get the gain factor G; Now the comparing the AR model with the AR equation we to compare that. So, we take the t equation side by side both can be written y n is there, in the left hand side for the AR model right hand side we are getting the previous the values of the output and G times the present input x n.

For the error equation what we are getting in the right hand side, the error term minus actually again the summation of the previous terms multiplied by the corresponding AR coefficient ok. So, now if we compare these two, what we get that the error term is nothing, but G times the present input signal.

(Refer Slide Time: 22:55)



Modelling of Biomedical Systems

Computation of gain factor G

It means the input signal is proportional to error signal, irrespective of nature of input.

When output error energy and H(z) are fixed, the total energy of the input signal is same as the total energy of the output signal.

Therefore,

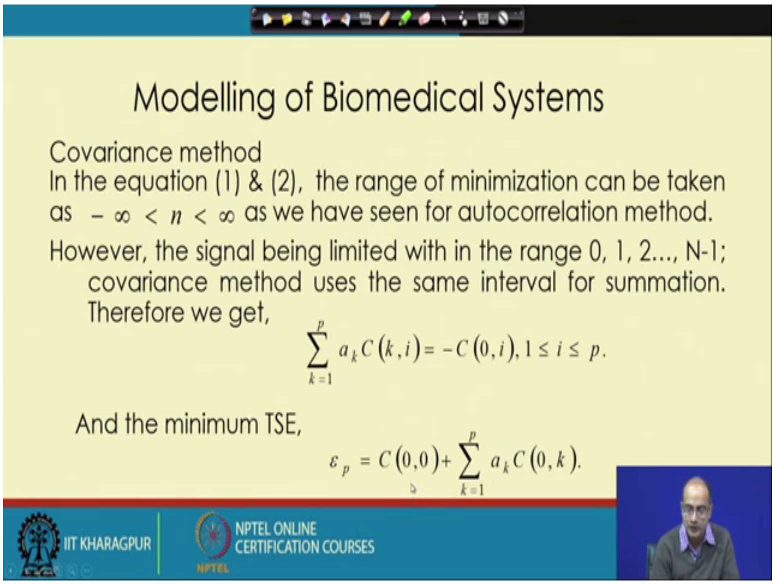
$$G^2 = \varepsilon_p = \phi_y(0) + \sum_{k=1}^{p} a_k \phi_y(k).$$

So, what we can do that we can get that input signal is proportional to the error signal or we can say other way round the error signal is proportional to the input, irrespective of the nature of input.

So, when we look at the output energy and that is for a fixed H z; that means, H z is computed, we need to have a energy balance in both the size in the input and the output because of the preservation of the energy. So, looking at that what we can get the G square is equal to that the prediction error for phi y equal to 0; that means, what we are suggesting that the input, if it is of say you need energy, then the solar energy of the driving is because of the scaling factor G.

So, that is same as the output error and we already know the expression of the output error. So, we can get G unique up to scaling the scaling factor is the scaling of the input signal. So, the simplest thing we can do we can assume, the driving signal energy is 1 and for that what is the value of the G we can get from the energy of the that error of prediction.

(Refer Slide Time: 25:07)



Modelling of Biomedical Systems

Covariance method

In the equation (1) & (2), the range of minimization can be taken as $-\infty < n < \infty$ as we have seen for autocorrelation method.

However, the signal being limited with in the range 0, 1, 2..., N-1; covariance method uses the same interval for summation. Therefore we get,

$$\sum_{k=1}^{p} a_k C(k,i) = -C(0,i), 1 \le i \le p.$$

And the minimum TSE,

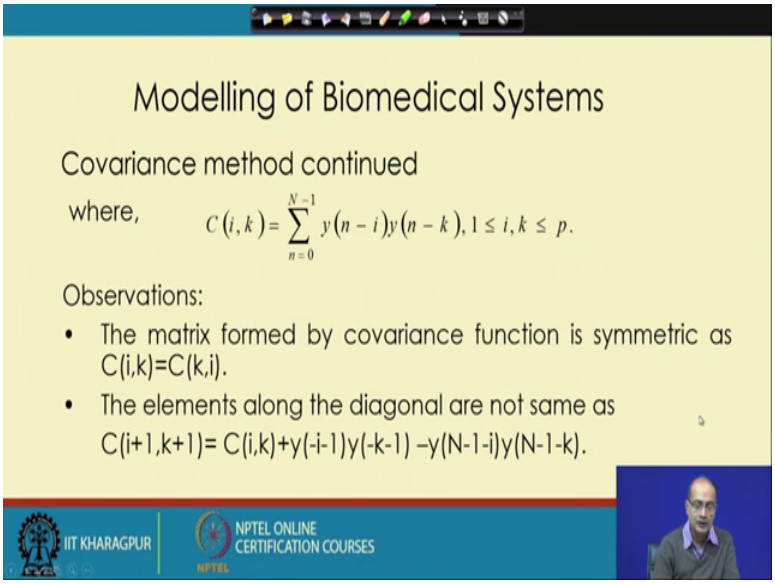
$$\varepsilon_p = C(0,0)+\sum_{k=1}^{p} a_k C(0,k).$$

So, with that we complete one part. Next we look at a different technique of computing the actually the AR coefficients. This new technique it is called covariance method, the previous technique as we are using the autocorrelation, we can call them as the autocorrelation method for computing the AR coefficients.

So, in this covariance method again, we use the same equation 1 and 2 the only change occurs in the range of minimisation. The previous technique we have taken the range is minus infinity to plus infinity and; however, that what limited because of the limited number of sample, but here we start with the fact that, we have signals from 0 from index 0 to capital N minus 1 ok.

And over these that if we minimise that error, we get a little different form, in this case we get the equations in this way using that autocorrelation this coefficients a ks and we get a new term called C C k minus k comma 1, or k comma i in this form. So, and this C C capital Cs are the autocorrelation function will get that, but before that we look at the form of total squared error, again it is expressed in terms of the covariance function C, which is replacing the autocorrelation function used in the previous autocorrelation method.

(Refer Slide Time: 27:30)



Modelling of Biomedical Systems

Covariance method continued

where, $$C(i,k)=\sum_{n=0}^{N-1} y(n-i)y(n-k), 1 \leq i,k \leq p.$$

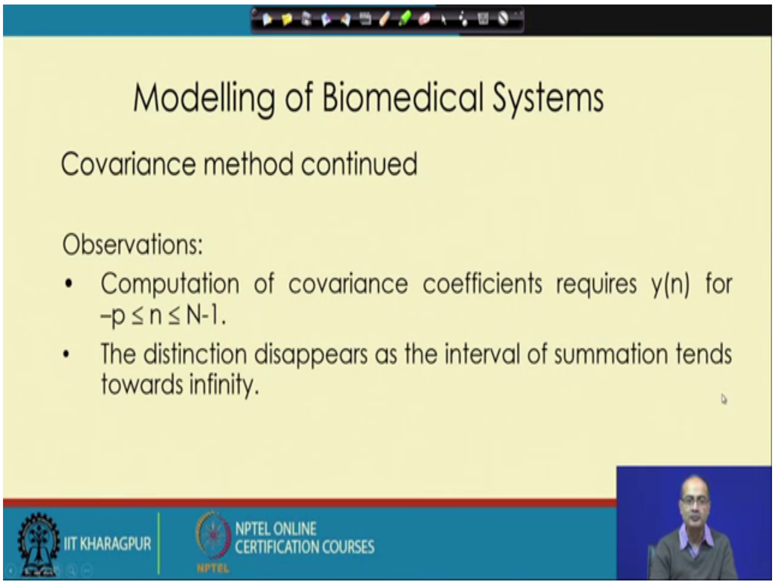
Observations:

- The matrix formed by covariance function is symmetric as C(i,k)=C(k,i).
- The elements along the diagonal are not same as C(i+1,k+1)= C(i,k)+y(-i-1)y(-k-1) –y(N-1-i)y(N-1-k).

And the form of that C i comma k is very similar to the previous one. However there is some differences that we observe, that matrix form by the covariance function it is again what we are getting it is symmetric because, C i comma k and C k comma i it is same because the terms the product is a commutative operation; So, the value as same if you change the order.

So, again we get a symmetric matrix, but everything is not same, the element along the diagonal are not same, that C 0 0 C 1 1 C 2 2, they are not same ok, there is a change and here we are showing that change in a quantitative way, if I look at C i comma i plus 1 comma k plus 1 in a more general sense, it is using that ci comma k plus, we are having a new term and we need to eliminate one term out of ci comma k. So, there is a change or we can say that the summation window is shifted by one place ok. So, that creates the difference we do not get any more that to a please matrix.

(Refer Slide Time: 29:33)



Modelling of Biomedical Systems

Covariance method continued

Observations:

- Computation of covariance coefficients requires y(n) for $-p \leq n \leq N-1$.
- The distinction disappears as the interval of summation tends towards infinity.

So, what we get instead, the computation of the covariance coefficients for that, we need the data y n, we need it from minus p 2 n minus 1 for that range. So, we have to if the data is given from 0 to or 1 to n we need to Reindex in it, and index it in this way. So, that we have all the values available the output signal is available from the lag, or from the instance minus p to capital n minus 1 ok.

However that different that is created between the autocorrelation and auto covariance method, this distinction disappears at the interval of summation tends to infinity, if we have more and more data, that difference actually disappears which is caused by addition and the deletion of one sample from two different sides ok. So, they actually go to the same thing. So, that is about the auto covariance method and we stop here for this session.

Thank you.